

1/6

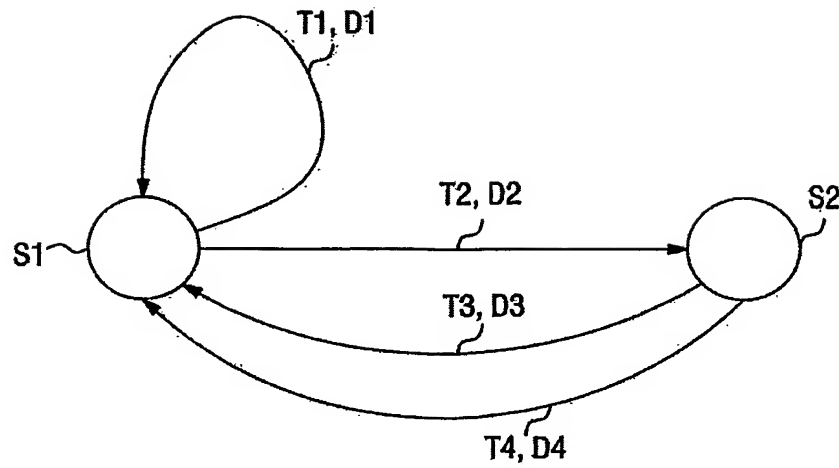


FIG.1
(Prior art)

AR \rightarrow <command> = <play> | <stop> | <goto>;
 AR \rightarrow <play> = (play|go|start) {PLAY};
 AR \rightarrow <stop> = (stop|halt|quit) {STOP};
 KR \rightarrow <goto> = go to line <lineno> {GOTO};
 AR \rightarrow <lineno> = 1 {ONE} | 2 {TWO} | 3 {THREE};

} \leftarrow GR

FIG.2
(Prior art)

2/6

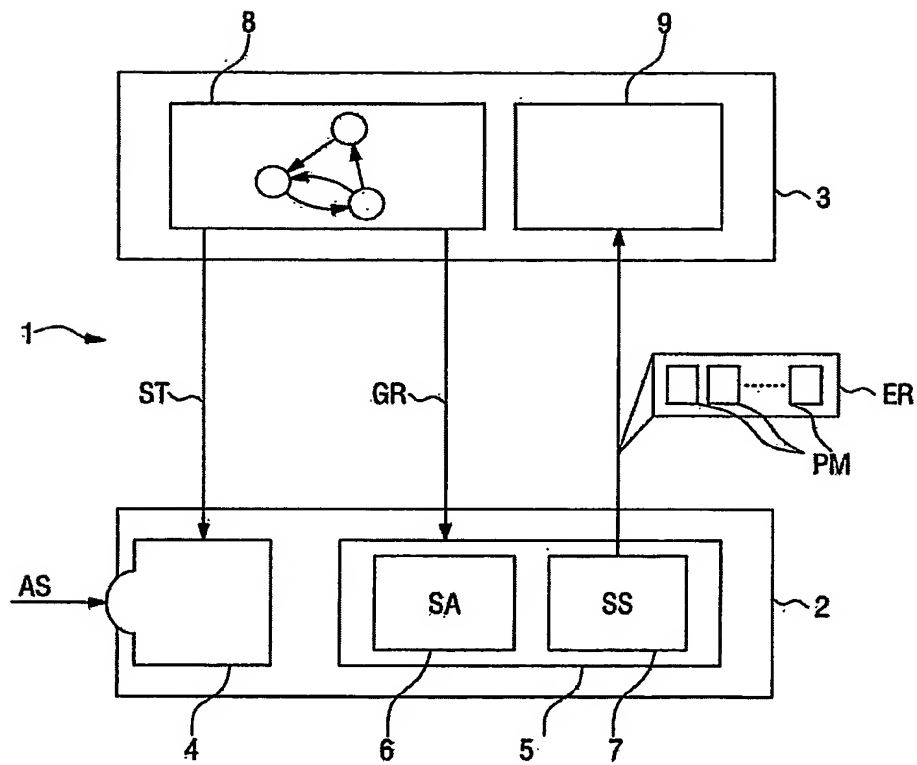


FIG.3

3/6

BK { abstract class Grammar {
 PhraseGrid GetPhraseGrid()
 object Value(Phrase phrase)
 abstract ParseNode[] PartialParse(Phrase phrase, int start)
}

SR { class PhraseGrammar : Grammar {
 PhraseGrammar(Phrase grammarphrase)
 ParseNode[] PartialParse(Phrase phrase, int start)
}

AR { class ChoiceGrammar : Grammar {
 ChoiceGrammar(param Grammar[] alternatives)
 ParseNode[] PartialParse(Phrase phrase, int start)
}


KR { class ConcatenatedGrammar : Grammar {
 ConcatenatedGrammar(param Grammar[] constituents)
 ParseNode[] PartialParse(Phrase phrase, int start)
}

GK {

FIG.4a

4/6

```
AR→ Grammar command = new ChoiceGrammar(play, stop, goto)
AR→ Grammar play = new ChoiceGrammar(
    new PhraseGrammar("play"),
    new PhraseGrammar("go"),
    new PhraseGrammar("start")
)
AR→ Grammar stop = new ChoiceGrammar(
    new PhraseGrammar("stop"),
    new PhraseGrammar("halt"),
    new PhraseGrammar("quit running")
)
AR→ Grammar lineno = new ChoiceGrammar(
    new PhraseGrammar("1"),
    new PhraseGrammar("2"),
    new PhraseGrammar("3")
)
KR→ Grammar goto = new ConcatenatedGrammar(
    new PhraseGrammar("go to line"),
    lineno
)
```



GO

FIG.4b

5/6

```
KR → Grammar multiplication = new ConcatenatedGrammar(  
    new NumberGrammar(1,9),  
    new PhraseGrammar("times"),  
    new NumberGrammar(1,9)  
)
```

} GO
↓

```
SE → void OnSynthesize(SynthesizeEventArgs e) {  
    e.Value = e.Values[0] * e.Values[2]  
}
```

FIG.5

6/6

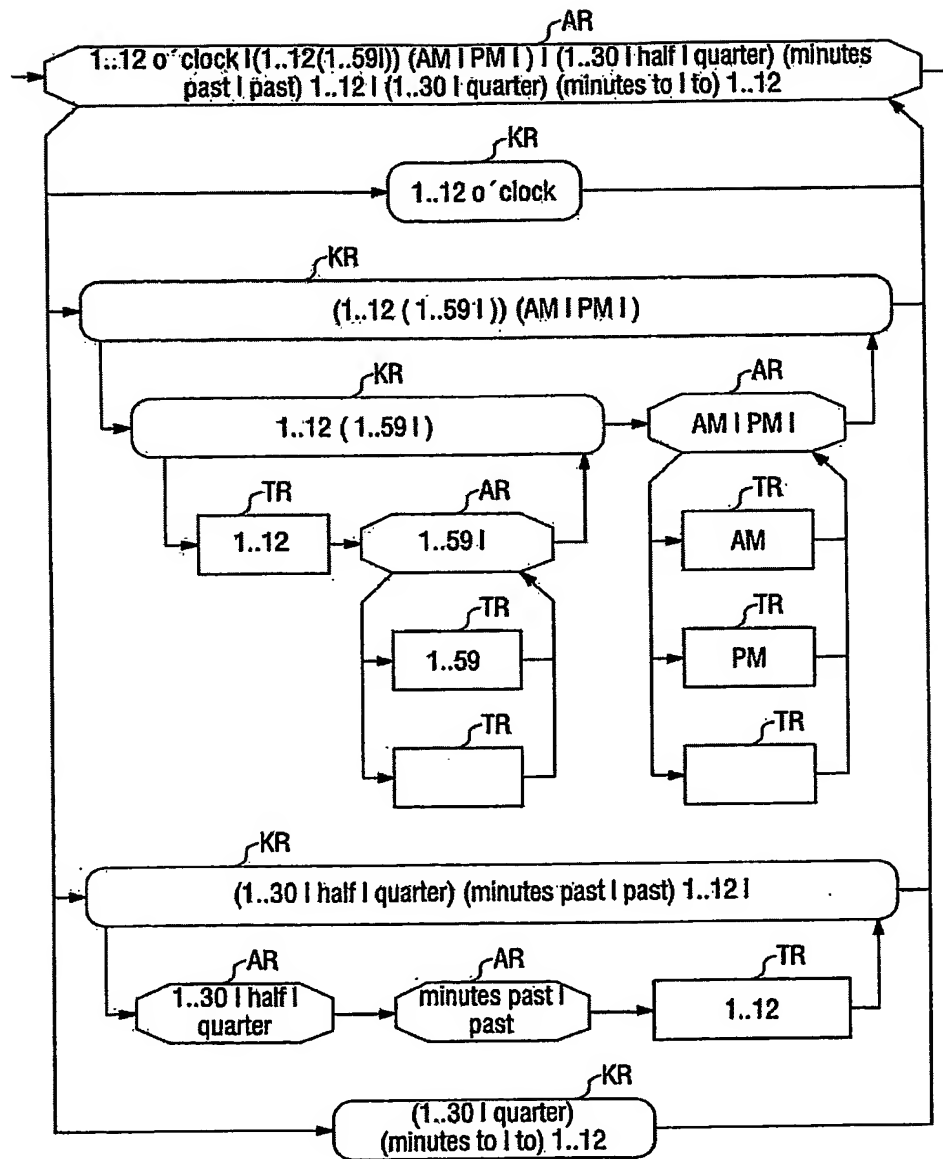


FIG. 6